

REMARKS

We have amended the independent claims to make more clear the scope that was implicit in the original claim language. That is, these changes are only to form and not to substance.

The Examiner continues to reject the claims 1-31 under 35 U.S.C. §103(a) as being obvious in view of Carusone (U.S. 5,157,667) over Reynolds (U.S. 6,138,161). We continue to believe, for many of the reasons presented by us before and for additional reasons presented below, that the Examiner's rejection is unsupported by the references.

However, before discussing what is missing from the Carusone system, we want to point out a fundamental difference in the two approaches. To perform fault isolation, Carusone relies on reports sent to a central location by each of the network units that detects the fault. In contrast, to identify a failed unit, the described embodiment in the present application relies on communications initiated by the central location. That is, in Carusone's system the network units distributed throughout the network initiate failure reporting; whereas, in the described embodiment, the central location "pings" the network units.

In Carusone's system, each of the network units includes a link adapter for each link to which it is connected. Each network unit stores in its local memory what Carusone refers to as "nearest neighbor" information:

...each unit of a pair of link coupled units, initially or on reconnection, interrogates a link adapter at the other end of the link for an identifier that identifies both the remote unit and the remote link adapter. This "nearest neighbor" information is stored locally at each unit, and is transmitted to the central location when an error is detected. (Abstract)

Upon detecting a failure, the network unit sends a failure report to the central location:

...according to the invention, whenever a failure occurs, failure reports are sent by each unit that observes the failure, to a central location. Each failure report includes the LAID of the link adapter that detected the failure as well as the LAID of the link adapter at the other end of the link (the previously stored LAID of the nearest neighbor). (Col. 5, lines 19-25).

...each LAID is basically a unit ID plus a unique number indicating a specific adapter on that unit (Col. 8, line 64 to col. 9, line 2).

The central location collects the received failure reports and then analyzes them to isolate the source of the problem.

In the described embodiment, a host computer 2 periodically "pings" each of the units on the network by sending a packet to that unit. If the target unit does not respond to the ping, thus indicating a failed attempt to communicate, the host computer identifies the neighbors of the target unit. Then, the host computer pings each of the neighbors, in turn, to determine if any of them are active. If any one of them is active, the host computer knows that the target unit is a failed unit, i.e., is the root cause of the failed attempt to communicate with it. If none of the neighbors is active, then the host computer concludes that the target unit is not the root cause of the failed attempt to communicate. The host computer then selects one of the inactive neighbors as a new target unit and pings each of its neighbors. This process continues until the host computer finds a target unit that has an active neighbor, at which point the host computer identifies that target unit as the root cause of the failed attempts to communicate. (see pages 8-10 of the specification).

The Examiner recognizes that, like the claimed invention, Carusone also uses neighbor or nearest neighbor information. The claimed invention, however, does not and is not intended to cover the general concept of using nearest neighbor information to troubleshoot network problems. Rather, it is directed to the more limited concept of using neighbor information to identify targets to ping so as to isolate the failed device. Though both systems use neighbor information, they use that information in quite different ways. Carusone uses neighbor information to analyze and interpret the meaning of the failure reports that are collected by the central location. The claimed invention uses the neighbor information to identify the targets to ping in order to isolate the root cause of a device failure.

We now turn to the specific differences between what is claimed and what is taught by the two references upon which the Examiner relies. With regard to claim 1, we submit that there is no support whatsoever in Carusone for concluding that if an attempt to communicate with a target device fails that there is any entity within the Carusone system that then performs the function of "determining if the target device has an active neighbor." Determining if the target device has an active neighbor simply plays no role whatsoever in Carusone's technique for isolating the fault. Furthermore, there is absolutely no support for characterizing the Carusone system as "identifying the target device as a failed device" in response to finding that the target device has an active neighbor.

To see that what we have just argued is indeed correct, it is worthwhile to try to map claim 1 onto a hypothetical system that operates in accordance with the teachings of Carusone. For our hypothetical system, assume that there are multiple units on a network, including unit A, unit B, which is a neighbor to unit A, and unit C, which is a computer at a "central location." Also assume that unit A attempts to communicate with neighboring unit B and that attempt fails. According to Carusone, unit A will then send a fault report to entity C. The fault report will at least include the LAID (link adapter ID) for unit A and for unit B, as well as a brief description of the fault that was detected. All of the other entities on the network that "see" the same fault will also report back to unit C in a similar manner. After all of the generated the fault reports are received, unit C analyzes them and identifies the nature and cause of the fault.

On way of attempting to map the claim onto this hypothetical system might be to view unit B as the "target device." In that case, unit A's attempt to communicate with unit B might correspond to "attempting to communicate with a target device," as required by claim 1. Under this approach, however, there will not be any subsequent operation by any entity that involves "determining if the target device has an active neighbor," as is also required by claim 1. Whether the target device has an active neighbor is not an inquiry or a test that any element in the Carusone system makes. Indeed, the Carusone system does not check to see whether any active neighbors even exist. Rather, it simply gathers all failure reports and analyzes them to determine what the fault is and where it is located.

Among the reports the central location receives, there will likely be reports from nearest neighbors of unit B. But simply receiving failure reports sent by active neighbors of unit B certainly is not the same as or equivalent to "determining if the target device has an active neighbor." Furthermore, even if we assumed that it was the same or equivalent, no entity in Carusone's system, upon determining that the target device has an active neighbor, responds by "identifying the target device as a failed device," as is also required by claim 1.

Moreover, unit A's generation of a failure report about the failed communication with unit B means that unit B does have an active neighbor, namely, unit A. If the test of claim 1 were implemented by Carusone (i.e., "if it is determined that the target device has an active neighbor, identifying the target device as a failed device), then every report of a failed

communication would result in the system identifying the "target device" as a failed device, which obviously would not be true in Carusone's system.

Another way of attempting to map claim 1 onto the Carusone system might be to view the central location as an active fault report generator. Thus, for example, assume that unit C attempts to communicate with unit B and that attempt fails, thereby potentially satisfying the first element of claim 1. Then, presumably unit C will generate a fault report to itself regarding its failure in attempting to communicate with unit B. But again, even under this scenario, nothing in the Carusone system then attempts to determine whether there is an active neighbor of entity B and nothing in the Carusone system identifies unit B as a failed device on the basis of finding that it has an active neighbor.

The Examiner also relies on Reynolds to supply that which is missing from Carusone. But nothing in Reynolds supplies the above-identified features that are missing from Carusone.

We note that claim 1 is not specific as to what entity is performing each of the individual operations recited in the claim. And indeed, according to the wording of claim 1, the recited functionality could be distributed among multiple entities. Claims 12, 21, and 30, however, do require that the functionality be implemented in a single entity. In that case, it should be even more apparent that Carusone clearly does not teach or even suggest a single entity that has the troubleshooting functionality specified by claim 1.

For the reasons stated above, we believe that the claims are allowable and therefore ask the Examiner to allow them to issue.

Attached is a marked-up version of the changes being made by the current amendment.

Please charge our Deposit Account No. 08-0219 the required fees of \$400 for the Petition for Two-Month Extension of Time under 37 CFR 1.136(a), and \$320 for the Notice of Appeal under 37 CFR 1.17(b).


Applicant : Ferdinand Engel
Serial No. : 09/401,874
Filed : September 23, 1999
Page : 11

Attorney's Docket No.: 113794.124 US1

Please apply any other charges or credits to Deposit Account No. 08-0219.

Date: August 19, 2002

Respectfully submitted,



Eric L. Prah
Reg. No. 32,590

Hale and Dorr LLP
60 State Street
Boston, Massachusetts 02109
Telephone: (617) 526-6000
Facsimile: (617) 526-5000

1485203_1.doc

Version with markings to show changes made

In the claims:

Claims 1-31 have been amended as follows:

1. (Once Amended) A method of [identifying a failed device in] troubleshooting a network that includes a plurality of devices, said method comprising:
attempting to communicate with a target device among the plurality of devices;
[determining if the target device has an active neighbor] if the attempt to communicate with the target device fails, determining if the target device has an active neighbor; and
[identifying the target device as a failed device] if it is determined that the target device has an active neighbor, identifying the target device as a failed device.
2. (Once Amended) [A] The method according to claim 1, wherein the [attempting comprises sending a packet to the target device and waiting for a response from the target device] method is implemented by a computer on the network.
3. (Once Amended) [A] The method according to claim [1] 2, wherein[: the] determining if the target device has an active neighbor comprises:
identifying a neighbor of the target device;
attempting to communicate with [a] the identified neighbor of the target device; and
[the neighbor is determined to be active] if the attempt to communicate with the identified neighbor is successful, concluding that the identified neighbor is active.
4. (Once Amended) [A] The method according to claim [1] 2, further comprising locating a neighbor of the target device.
5. (Once Amended) [A] The method according to claim 4, wherein the locating comprises:
generating a neighbor table for the network; and
consulting the neighbor table to locate the neighbor of the target device.

6. (Once Amended) [A] The method according to claim 5, wherein the generating comprises:

polling the target device;
receiving a response from the target device; and
constructing the neighbor table based on the response.

7. (Once Amended) [A] The method according to claim 6, wherein:
the polling is performed periodically; and
the method further comprises updating the neighbor table based on the periodic polling.

8. (Once Amended) [A] The method according to claim 6, wherein:
the response comprises a network address of the neighbor; and
the neighbor table indexes the target device to the network address of the neighbor.

9. (Once Amended) [A] The method according to claim 8, wherein the target device:
stores a Management Information Base (MIB) II table containing the network address of the neighbor; and
prepares the response based on the MIB II table.

10. (Once Amended) [A] The method according to claim [1] 2, wherein the target device comprises a router or a switch, and the neighbor comprises a router, a switch, or a computer.

11. (Once Amended) A method of [identifying a failed device in] troubleshooting a network that includes a plurality of devices, said method comprising:

receiving information from the plurality of devices;
generating a neighbor table for the network [devices] based on the information provided [from] by the plurality of devices; and

[sending a packet to] attempting to communicate with a target device among the plurality of devices to determine if the target device is active;

wherein, if the target device is determined to be not active, the method further comprises:

[locating a neighbor of the target device] using the neighbor table to identify a neighbor of the target device;

[sending a packet to] attempting to communicate with the identified neighbor to determine if the identified neighbor is active; and

[identifying the target device as a failed device] if the identified neighbor is determined to be active, identifying the target device as a failed device.

12. (Once Amended) An apparatus for [identifying a failed device in] troubleshooting a network that includes a plurality of devices, said apparatus comprising:

a host computer including a processor; and

a memory which stores executable code[; and a processor] which when executed by said processor causes said host computer to [executes code] (i) [to] attempt to communicate with a target device among the plurality of devices, (ii) [to determine if the target device has an active neighbor] if the attempt to communicate with the target device fails, determine if the target device has an active neighbor, and (iii) [to identify the target device as a failed device] if the target device [has] is determined to have an active neighbor, identify the target device as a failed device.

13. (Once Amended) [An] The apparatus according to claim 12, wherein the processor attempts to communicate with the target device by sending a packet to the target device and waiting for a response from the target device.

14. (Once Amended) [An] The apparatus according to claim 12, wherein:
the processor determines if the target device has an active neighbor by attempting to communicate with a neighbor of the target device; and
the neighbor is determined to be active if the attempt to communicate is successful.

15. (Once Amended) [An] The apparatus according to claim 12, wherein the processor executes code to locate a neighbor of the target device.

16. (Once Amended) [An] The apparatus according to claim 15, wherein the processor locates the neighbor by:

generating a neighbor table for the network; and
consulting the neighbor table.

17. (Once Amended) [An] The apparatus according to claim 16, wherein the processor generates the neighbor table by:

polling the target device;
receiving a response from the target device; and
constructing the neighbor table based on the response.

18. (Once Amended) [An] The apparatus according to claim 17, wherein the processor performs the polling periodically and updates the neighbor table based on the periodic polling.

19. (Once Amended) [An] The apparatus according to claim 17, wherein:
the response comprises a network address of the neighbor; and
the neighbor table indexes the target device to the network address of the neighbor.

20. (Once Amended) [An] The apparatus according to claim 12, wherein the target device comprises a router or a switch, and the neighbor comprises a router or a switch.

21. (Once Amended) A computer program stored in a computer-readable medium, said program for troubleshooting [to identify a failed device in] a network that includes a plurality of devices, said program comprising:

code [to] for attempting to communicate with a target device among the plurality of devices;

code [to determine] for determining if the target device has an active neighbor if an attempted communication with the target device fails; and

code [to identify] for identifying the target device as a failed device if the target device has an active neighbor.

22. (Once Amended) [A] The computer program according to claim 21, wherein the attempting code sends a packet to the target device and waits for a response from the target device.

23. (Once Amended) [A] The computer program according to claim 21, wherein:
the determining code attempts to communicate with a neighbor of the target device; and
the neighbor is determined to be active if an attempted communication is successful.

24. (Once Amended) [A] The computer program according to claim 21, further comprising code to locate a neighbor of the target device.

25. (Once Amended) [A] The computer program according to claim 24, wherein the locating code comprises:

code to generate a neighbor table for the network; and
code to consult the neighbor table to locate the neighbor of the target device.

26. (Once Amended) [A] The computer program according to claim 25, wherein the generating code comprises:

code to poll the target device;
code to receive a response from the target device; and
code to construct the neighbor table based on the response.

27. (Once Amended) [A] The computer program according to claim 26, wherein:
the polling code performs the polling performed periodically; and
the computer program further comprises code to update the neighbor table based on the periodic polling.

28. (Once Amended) [A] The computer program according to claim 26, wherein:
the response comprises a network address of the neighbor; and
the neighbor table indexes the target device to the network address of the neighbor.

29. (Once Amended) [A] The computer program according to claim 21, wherein the target device comprises a router or a switch, and the neighbor comprises a router or a switch.

30. (Once Amended) A network system comprising:

a first device;

a second device; and

a third device located in a path between the first device and the second device on a network; wherein the first device comprises:

a computer including a processor; and

a memory which stores executable code[; and a processor] which when executed by said processor causes said computer to [executes code] (i) [to] send a packet to the second device to determine if the second device is active, (ii) if the second device is not active, [to] send a packet to the third device to determine if the third device is active, and (iii) [to] if the third device is determined to be active, identify the second device as a failed device [if the third device is active].

31. (Once Amended) [A] The network system according to claim 30, wherein the first device comprises a computer, the second device comprises a switch or a router, and the third device comprises a switch or a router.